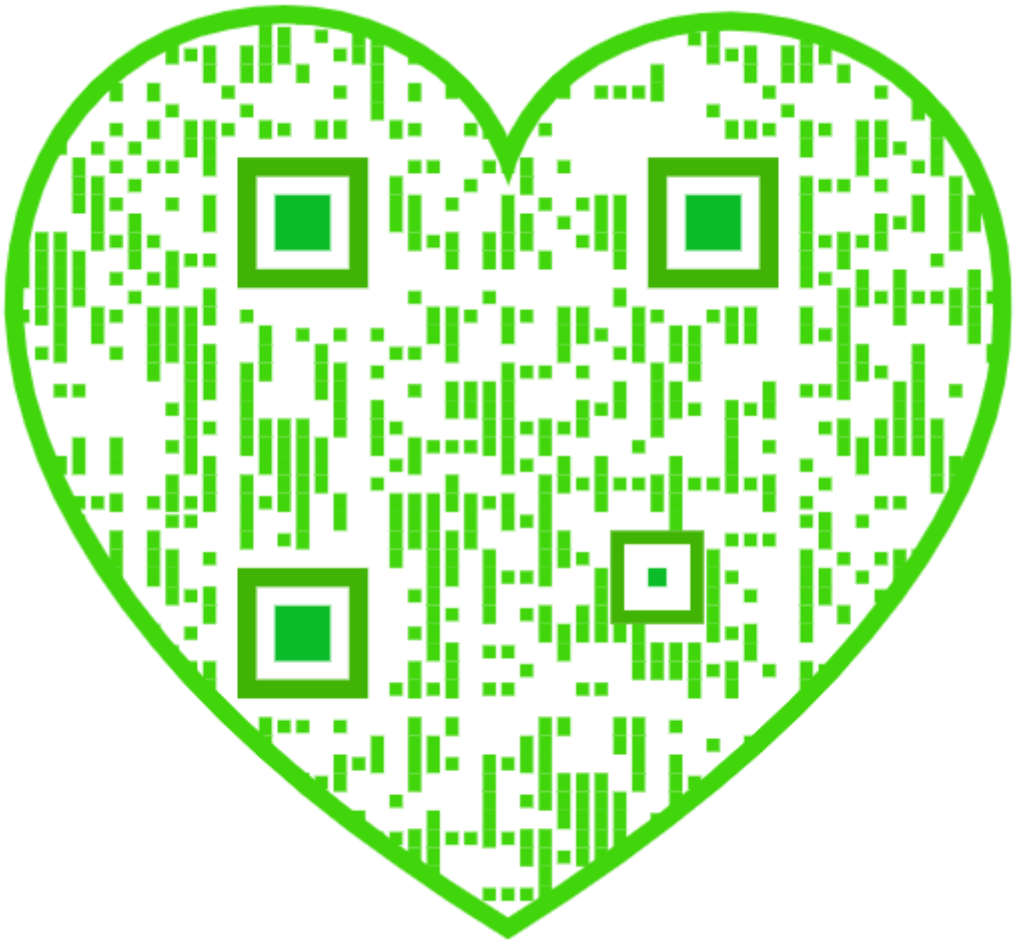
Master in Artificial Intelligence

Deployment III



Purpose

The purpose of the section is to help you learn how to deploy trained models into production environments to become a Successful Artificial Intelligence (AI) Engineer

At the end of this lecture, you will learn the following

- **An example of deploying trained models into production environments, ensuring they integrate smoothly with existing systems and meet performance requirements**

Example of how to deploy trained models into production environments

Trained Model

+

Existing Systems

Meet Performance Requirements

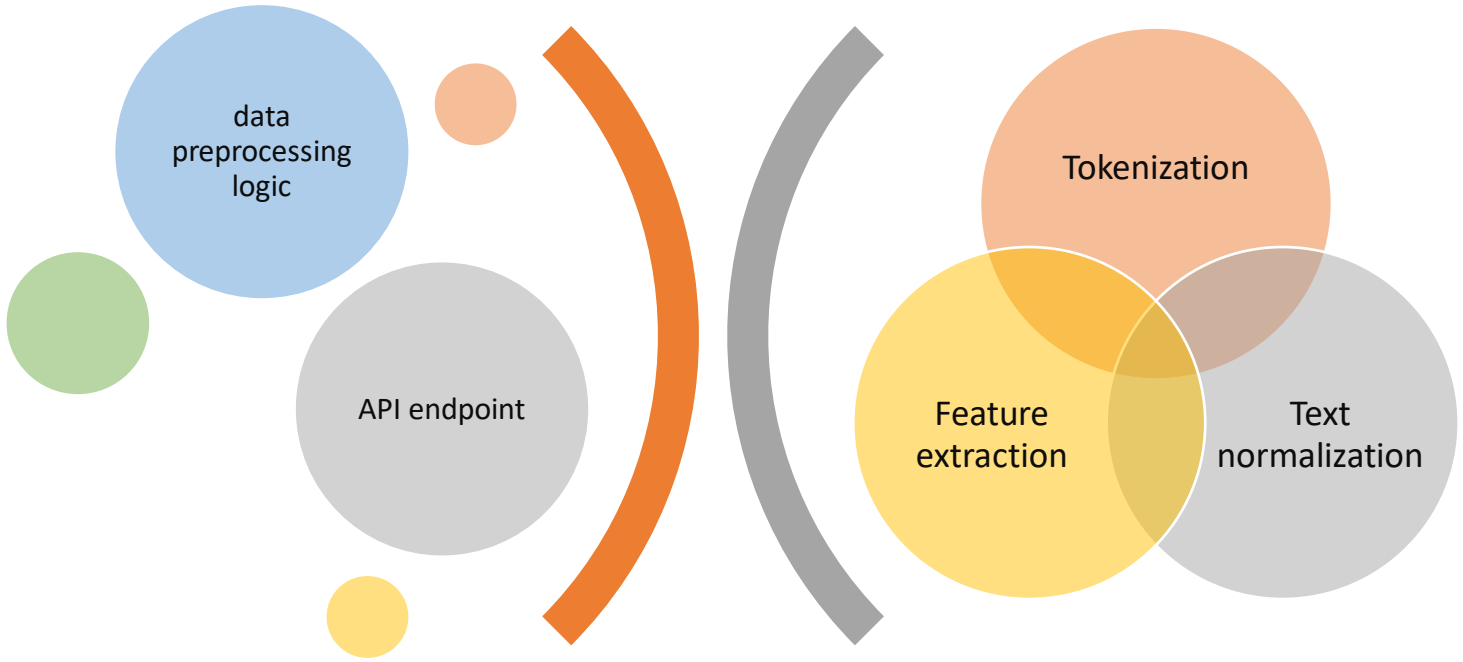
Serialization of Trained Model

```
import pickle

# Assuming 'model' is your trained sentiment analysis model
with open('sentiment_model.pkl', 'wb') as f:
    pickle.dump(model, f)
```

Setting up a RESTful API

```
from flask import Flask, request, jsonify

app = Flask(__name__)

@app.route('/predict_sentiment', methods=['POST'])
def predict_sentiment():
    # Get text input from request
    text = request.json['text']

    # Load the serialized model
    with open('sentiment_model.pkl', 'rb') as f:
        model = pickle.load(f)

    # Perform sentiment prediction
    prediction = model.predict([text])[0]

    # Return prediction as JSON response
    return jsonify({'sentiment': prediction})

if __name__ == '__main__':
    app.run(debug=True)
```

Input Data Preprocessing



Implemented

Handled text inputs

Performance Optimization

- Optimized the API endpoint for performance
  - Minimizing latency
  - Maximizing throughput
- Consider techniques such as
  - Caching
  - Batching
  - asynchronous processing

Integration with Existing Systems

Integrated

- Deployed API endpoint
- With existing systems or applications
- That require sentiment analysis functionality

Ensure compatibility with

- Data formats
- Protocols
- Authentication mechanisms used by other systems

Monitoring and Logging

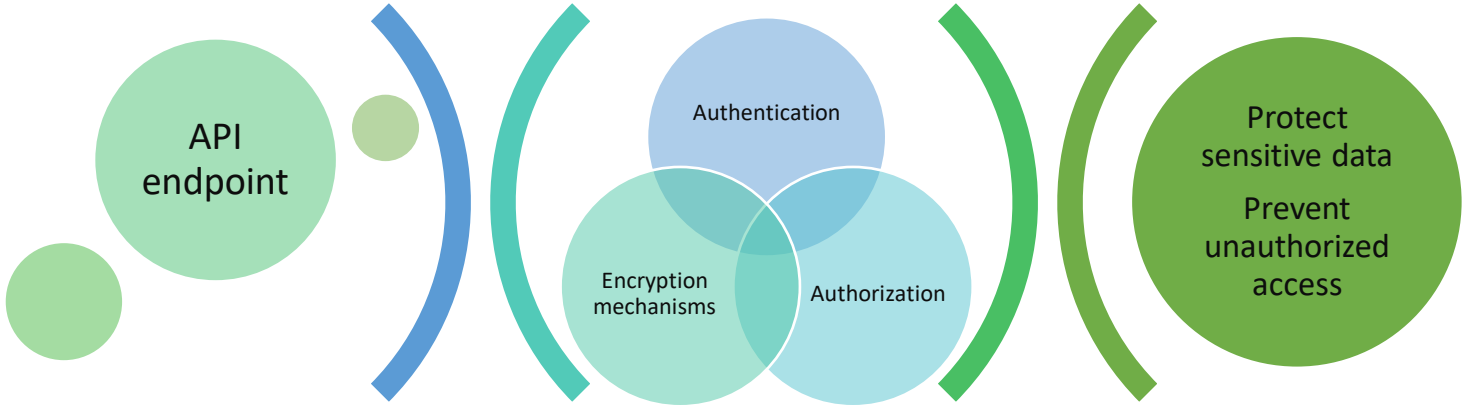
Implement logging and monitoring mechanisms to track

- API usage
- Performance metrics
- Errors in real-time

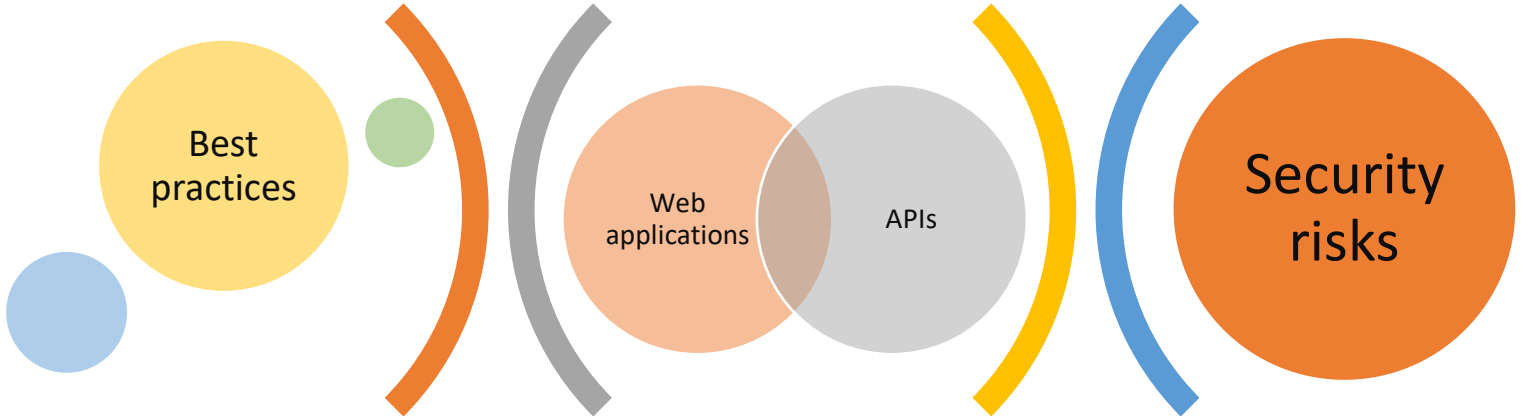
Monitor key metrics such as

- Response time
- Throughput
- Error rates

Security and Authorization



Secured Implemented To



Followed Securing Mitigate

Testing and Validation

Conducted

- Thorough testing and validation of the deployed API endpoint
- Ensured correctness, reliability, and performance under various conditions.

Performed

- Integration testing
- Load testing

Validated

- Behavior and performance of
- API in a production environment

Example of How to deploy trained models into production environments

Trained Model

+

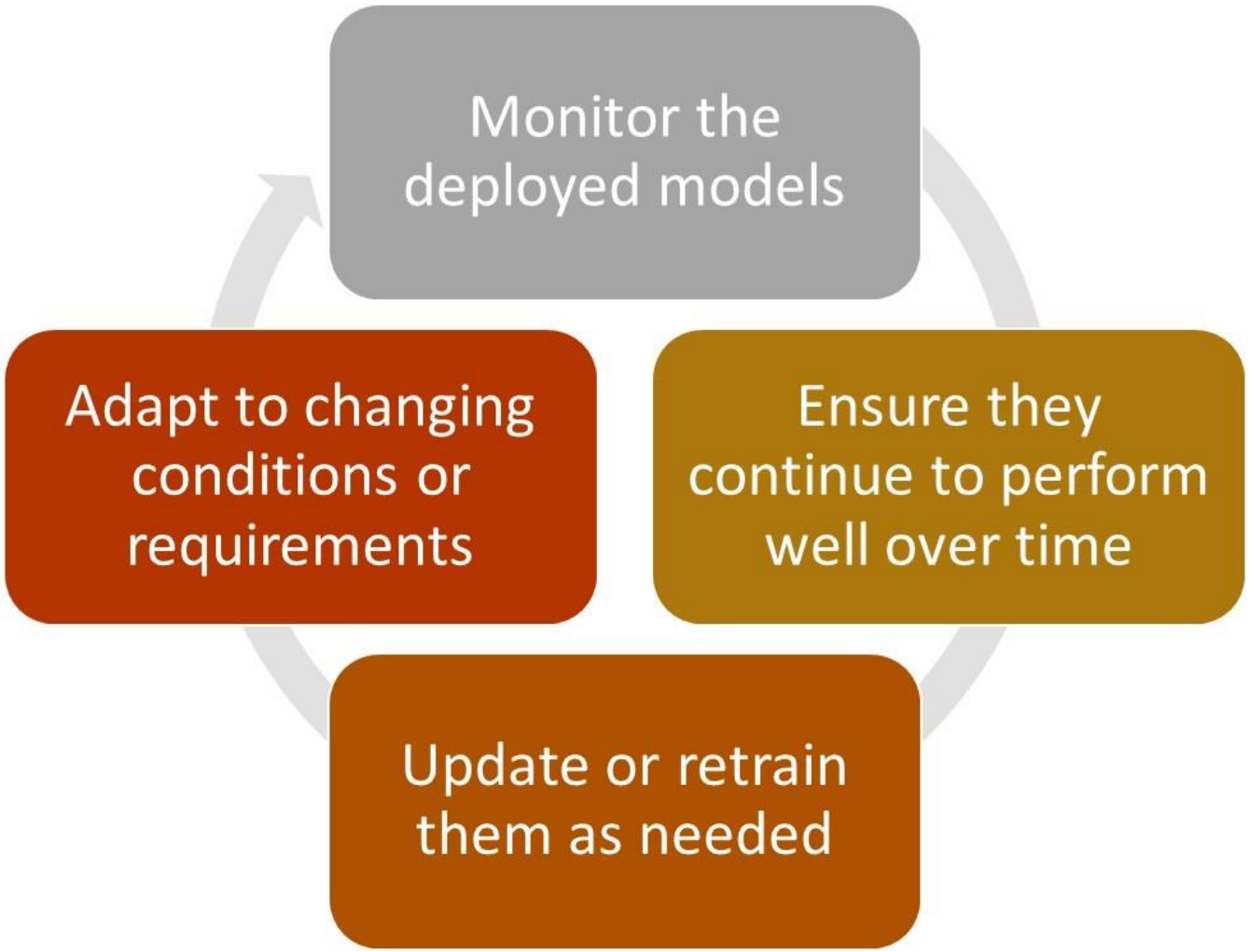
Existing Systems

→

Meet Performance Requirements

What is next?

Monitoring and Maintenance



Master in Artificial Intelligence

Deployment III